SENTIMENT ANALYSIS ON YOUTUBE COMENTS

A comparison based approach of performance analysis

The sentiment analysis is a powerful tool in the modern era solely dependent on information technology which lets us analyze data in a very intimate and detailed way. SA paves for one to understand the hidden connections between the action of a user and the information presented by him to the internet. This technology is also being used to predict user action based on the information given out by the user in the form of text. Thus the comments on various social platforms like Facebook, twitter, and on other online shopping websites like amazon reveal a lot of information on the user preference and understating of a topic or product. The implementation is done using python and the nltk(Natural Language Toolkit) library and the modules used in this project.

Normalization and Sorting

Normalizing a set of values is very important for comparative study and inference. In this particular project the normalized value of the sentiment score collected by using the tools defined in NLTK like VADER, Text Blob, Lexicon Polarity, Custom Classifier, is computed on the basis of the number of comment thread in each video. Thus a normalized of the sentiment score that takes care of the length of comment thread and also the length of the comment text in some context is of the utmost importance. This more so because if a video has more number of positive comment as individual sentences as compared to the number of comments being considered its value would otherwise be less if any other video which has equal or more number of positive comment text but has the number of comments double than the other video. In order to deal with scenario, we would have pave a way that generates normalized sentiment scores. The techniques used for normalizing various functions vary a little but the overall idea is the same.

Vader:

In Vader the normalization is done taking into consideration the length of the comment thread, that is the number of comments a video has or the maximum number that is being considered in this project. The equation used is given as

Thus the final output is between -1 to +1 also that negative sign of the values are taken care of and the normalized value can also be negative if the non-normalized value is negative.

Text Blog:

Similarly, like the Vader normalization technique we employ the squaring and square rooting on the non-normalized and the normalized values respectively. This technique though has a catch, unlike the VADER the sentiment score of individual comments is not between -1 and +1 here few of the results can be actually +1 and -1 which are not very promising values if the final value has to kept between these two outer limits. Thus for a negative sentiment score the equation can be,

Custom Classifier

The Normalization used for the custom designed sentiment classifier is very unique as this only returns a integer value of 1 and -1 or ‘pos’ and ‘neg’ as its output for the sentiment score of the individual comments. Thus the value should be normalized in-order for the program to generate a better response. The factors considered in this scenario is the length of each comment string and off course the total number of comments for each video. The equation for a ‘neg’ result translated to =1 is,

NLTK Lexicon Score

The sentiment score generated by this similar to Custom Classifer but the underlying functioning varies a lot as compared to the later. Due to this fact the normalization technique used in this has a similar mathematical structure. The equation for a ‘pos’ or +1 would have to be,

Results

The results generated by the program after using four different tools like VADER, TEXT BLOB, NLTK LEXICON ANALYZER, CUSTOM CLASSIFIER are quite impressive. The program as explained earlier perfoms sentiment analysis of the youtube comments set by a search keyword initially. In this example the search keyword used is ‘lamborgini huracan’ the favourite sports car. The table of the generated score data is as follows

The first table is a representation of the unarranged data as it is generated by the program. The second table are sorted on the basis of ‘compound’ which is an abbreviation of VADER compound score. This score reflects the polarity of the sentiment with a value for intensity. The third table is sorted on the basis of ‘lexicon\_score’ which an abbreviation of the NLTK lexicon analyzer score which in short gives the result on the basis of negative and positive words in the individual comments. The third table is sorted on the basis of the VADER`s negative intensity score abbreviated as ‘neg’.

The above table represents sentiment analysis score for individual score generated by the Custom Sentiment Classifier.

Conclusion

The concept of sentiment analysis on the YouTube comments to generate a more intuitive video recommender system is disruptive idea. The project performs as promised and gives out relatively god results and can also be used in practical applications.

Future Work

The design of the Custom Classifier is very intuitive and can be a little weak in terms of execution time. Thus better implementation techniques can be employed in order to improve the performance of the Custom classifier and in turn the whole project.

Reference

1. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, Steven Bird, Ewan Klein, Edward Loper
2. <http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf>
3. <http://blog.algorithmia.com/benchmarking-sentiment-analysis-algorithms/>